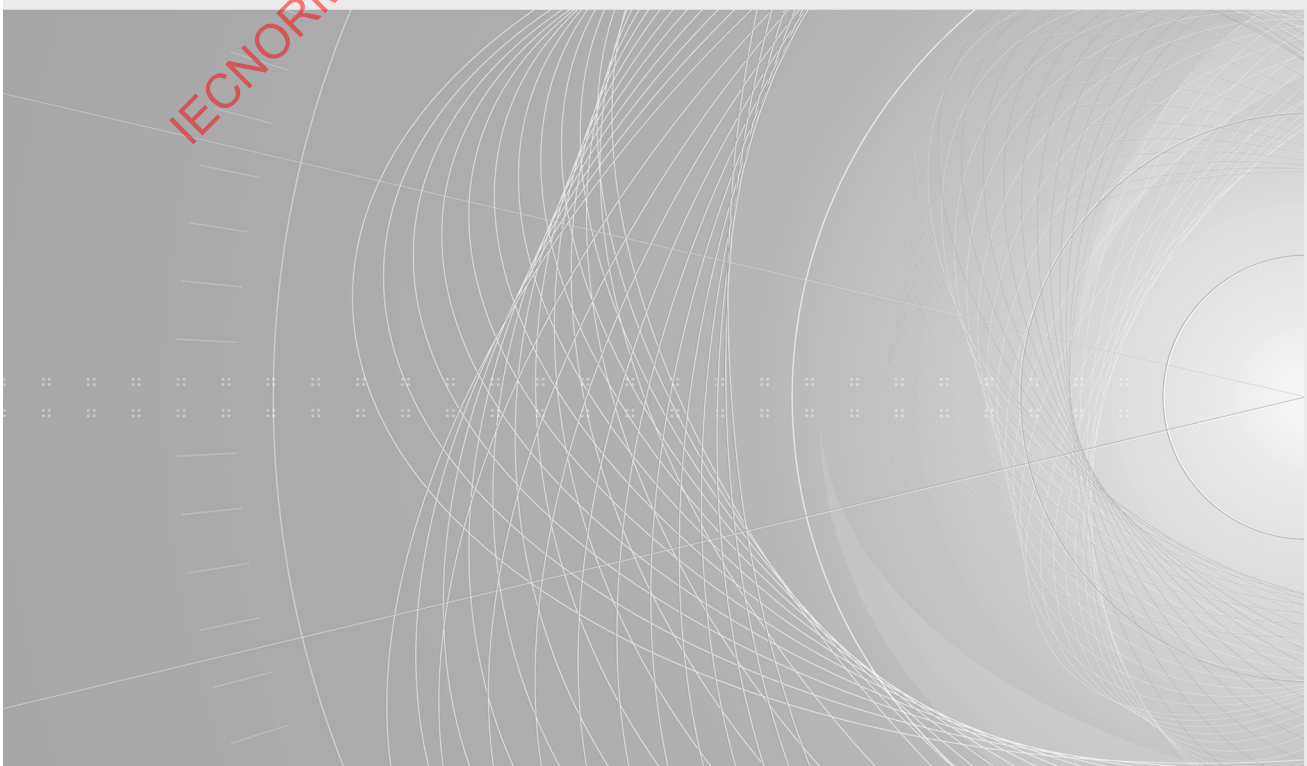
IEC 62264-6

Edition 1.0 2020-06

INTERNATIONAL STANDARD

colour inside

Enterprise-control system integration – Part 6: Messaging service model



IEC 62264-6:2020-06(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC 62264-6

Edition 1.0 2020-06

INTERNATIONAL STANDARD

Enterprise-control system integration – Part 6: Messaging service model

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ICS 25.040.40; 35.100.70

ISBN 978-2-8322-8453-7

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

ENTERPRISE-CONTROL SYSTEM INTEGRATION –

Part 6: Messaging service model

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62264-6 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee TC65: Industrial-process measurement, control and automation.

The text of this International Standard is based on the following documents:

FDIS	Report on voting
65E/706/FDIS	65E/724/RVD

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the IEC 62264, published under the general title *Enterprise-control system integration*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under "http://webstore.iec.ch" in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

This document is based on the use of IEC 62264 object models defined in IEC 62264-2, IEC 62264-4 and IEC 62264-5 to define a set of services that may be used to exchange messages. This document defines a messaging service model (MSM) for exchanging messages in a publish-subscribe mode and a request-response mode.

The Messaging Service Model provides a method for applications to send and receive messages from MSM service providers without regard to the underlying communication mechanism, as part of a complete application-to-application data exchange.

This document defines a model for message exchange services (Messaging Service Model) that are designed to provide a technology independent method for sending and receiving transaction messages to or from underlying exchange services.

The knowledge requirements to interface to just one message exchange system can be immense and are usually not transferable to a different system. MSM defines a single interface, independent of the underlying exchange services, for exchanging data objects defined by IEC 62264-2 and by IEC 62264-4. This removes the need for vendors to build custom interface after custom interface, and for end users to get locked into a single vendor because their investment prevents them from reusing any of the integration efforts.

Exchanging the data objects between different computer system applications involves multiple different steps, as shown in Figure 1.

a) The applications usually have different internal representations of exchanged objects in their own local data stores. This representation is usually converted from the local format to a commonly accepted global format. IEC 62264-2 defines models of a global format for Level 4-3 data exchanges. IEC 62264-4 defines models of a global format for Level 3-3 data exchanges. This conversion, from local to global and global to local, is usually performed twice for any two-way communications.

   EXAMPLE 1 Assume two applications, ALPHA and BETA: the ALPHA application initiates a data exchange with the BETA application, and BETA responds back to ALPHA. The format conversions are: ALPHA's local format to global format for the request data, global format to BETA's local format for the request data, BETA's local format to global format for the response data, and global format to ALPHA's format for the response data.

b) Conversion is performed to align the namespaces among the exchanging applications and is usually performed four times for any two-way communications.

   EXAMPLE 2 Names for elements of data can be codes, tag names, or equipment identifiers.

   EXAMPLE 3 Data which are represented in one element namespace, such as codes 1,2,3,4, can have a different namespace in another application, such as codes Ok, Done, Error, Delay.

c) Once information is in the global format with global names, the exchanged information is sent from one application to another application.

d) Messages are transported from one application to another, either within the same computer environment or across computers. Transport mechanisms are defined in other standards, such as TCP/IP and Ethernet standards.

e) When data exchange information is received, there are specific rules that define what resultant data are to be returned. The transaction rules are defined in IEC 62264-5.

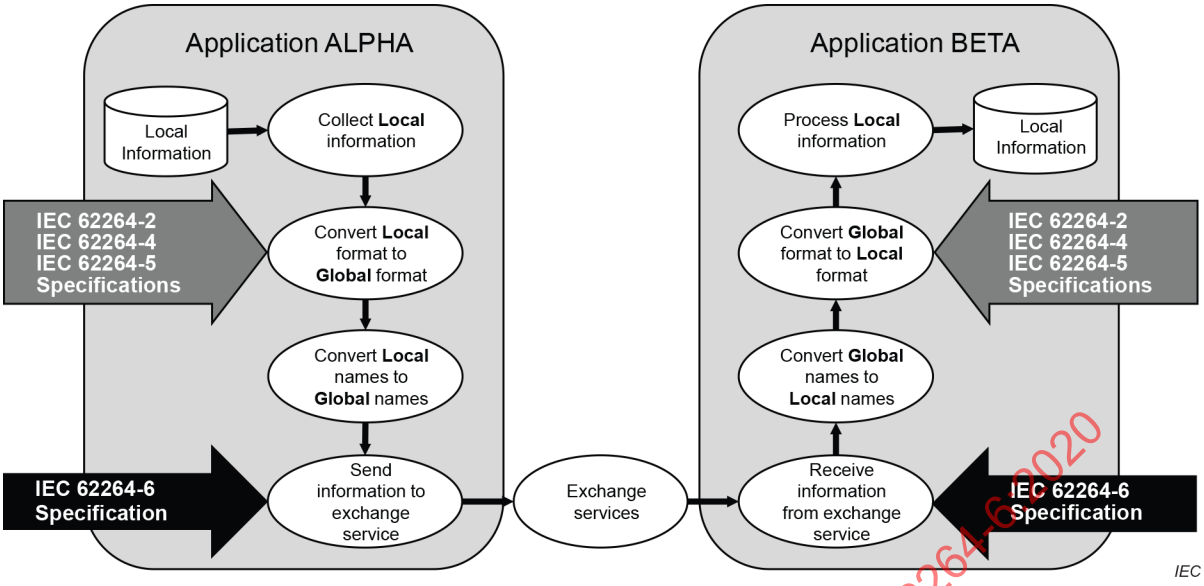


Figure 1 – Steps in application-to-application communication

MSM is a minimal interface subset that can reside on most exchange services and is based on well-defined and structured data objects and transaction messages.

Each layer shown in Figure 2 addresses a specific element of application data exchange.

1) A Data Object layer defines the meaning, format, and structure of the basic elements of exchanged information.

   NOTE 1 This layer uses application space specific definitions, such as the IEC 62264-2 and IEC 62264-4 object definitions, MESA B2MML, MIMOSA CCOM objects, and "Nouns" defined in OAGIS.

2) A Transaction layer defines the meaning, format, and structure of actions to be taken on the data objects.

   NOTE 2 This layer can use IEC 62264-5 transaction style specific definitions. Another transaction layer definition could be the OAGIS "Verb" definitions.

3) The MSM defines an interface to the OSI Application layer's services.

4) The application, presentation, session and lower level layers define the meaning, format, and structure for coordination, buffering, and exchange of messages or files. These layers contain transfer or exchange style specific definitions, such as Enterprise Service Buses, Enterprise Message Delivery Systems, the OPC UA specification (IEC 62541 standard), RSS, FTP, Named Pipes, Ethernet, TCP/IP, HTTP, and others.

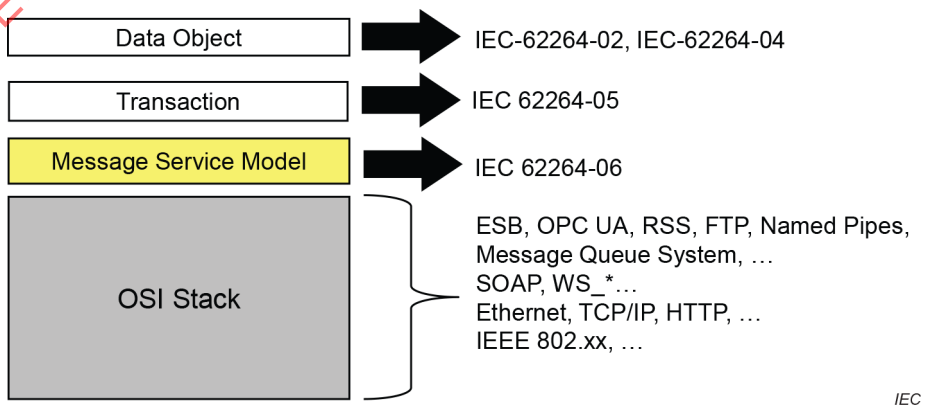


Figure 2 – Defined standards at each level

The IEC 62264-5 standard defines transactions on the information. The Messaging Service Model (MSM) defines an interface to methods for exchange. In a sense, MSM defines the standard "on-ramp" and "off-ramp" to the application layer services. It defines how data is placed into exchange methods and how it is retrieved from the exchange methods.

NOTE 1 Message synchronization using the MSM service is distinct from the message synchronization provided by the 62264-5 transaction models as well as distinct from the synchronization mechanisms provided at lower levels of the communications stack.

NOTE 2 In this document, asynchronous message exchanges between consumers and producers can be considered to be pairs of distinct, unidirectional messages.

This document includes two informative annexes. Annex A is informative. It provides considerations for (MSM) service providers. Annex B is informative. It provides a brief description of Enterprise Service Buses as a message exchange mechanism.

ENTERPRISE-CONTROL SYSTEM INTEGRATION –

Part 6: Messaging service model

1 Scope

This document defines a technology independent model for a set of abstract services that is located above the application layer of the OSI model, and that is used for exchanging transaction messages based on the transaction models defined in IEC 62264-5. The model, which is called the Messaging Service Model (MSM), is intended for interoperability between manufacturing operations domain applications and applications in other domains.

NOTE It is recognized that other sets of services not defined in accordance with this document are possible for the exchange of MOM information and are not deemed invalid as a result of this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62264-1, *Enterprise-control system integration – Part 1: Models and terminology*

IEC 62264-2, *Enterprise-control system integration – Part 2: Object and attributes for enterprise-control system integration*

IEC 62264-4, *Enterprise-control system integration – Part 4: Objects models attributes for manufacturing operations management integration*

IEC 62264-5, *Enterprise-control system integration – Part 5: Business to manufacturing transactions*

3 Terms, definitions, abbreviation, and conventions

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.1.1
channel description
text that describes a channel

3.1.2
channel type
primary use of a channel for publications or for requests

3.1.3
channel URI
primary identifier for a channel

3.1.4
filter expression
filtering element that may be applied to messages on a channel

3.1.5
listener identification
implementation defined element that is used to indicate to an application when a new message has arrived

3.1.6
message content
body of the message

3.1.7
message expiry
duration until the expiration of a publication message on a publication channel

3.1.8
message ID
identifier generated upon posting of a message to a channel in a session

3.1.9
MSM service
set of implemented services based on the messaging service model

3.1.10
MSM Service Provider
application that exposes and implements the MSM service

3.1.11
namespace
collection of names or words that define a formal and distinct set

3.1.12
security token
physical device or software code used to gain access to a channel

3.1.13
session ID
identifier generated upon an application creating a session on a channel and provided to the application for use in the MSM service

3.1.14
topic
identification of the information content in a message

3.2 Abbreviations

B2MML	Business to Manufacturing Markup Language
CB (radio)	Citizens' Band radio
CCOM	Common Conceptual Object Model
ERP	Enterprise Resource Planning

ESB	Enterprise Service Bus
FTP	File Transfer Protocol
HTTP	Hypertext Transmission Protocol
JMS	Java Message Service
MSM	Messaging Service Model
MIMOSA	Operations and Maintenance Information Open System Alliance
OAG	Open Applications Group
OAGIS	Open Applications Group Integration Specification
OMAC	Organization for Machine Automation and Control
OpenO&M	Open Operations and Maintenance Group
OPC UA	OPC Unified Architecture
REST	Representational State Transfer
RSS	Really Simple Syndication
SOAP	Simple Object Access Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
UDDI	Universal Description, Discovery and Integration
URI	Universal Resource Identifier
WS_*	World Wide Web Service standards
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations

3.3 Conventions

Input and returned parameters defined in Clause 6 are mandatory unless they are explicitly defined as optional.

4 Overview of the MSM

4.1 Positioning of the MSM

Application to application data exchange is represented in communication models as a single "Application" layer, as shown in Figure 1. Although the OSI model represents communication systems, additional mechanisms are required to describe the complexity of object-based application-to-application transactional communication. These mechanisms are specified by standards for defining data objects and for defining transaction messages

The generic data object models are defined by IEC 62264-2 and IEC 62264-4 and they are implemented by specific data representation technologies (e.g. B2MML, MIMOSA CCOM and OAGIS nouns). The generic transaction messages are defined by IEC 62264-5 and they are implemented by specific message transaction technologies (e.g. OAGIS 9.0 Verbs). The data objects are transformed into transaction messages. These transaction messages are passed to the underlying exchange services through the MSM service, which is based on the Message Service Model defined by this document. This message flow is shown in Figure 3.

MSM is an interface that resides on the exchange service used by an application and exchanges the transaction messages using the exchange services.

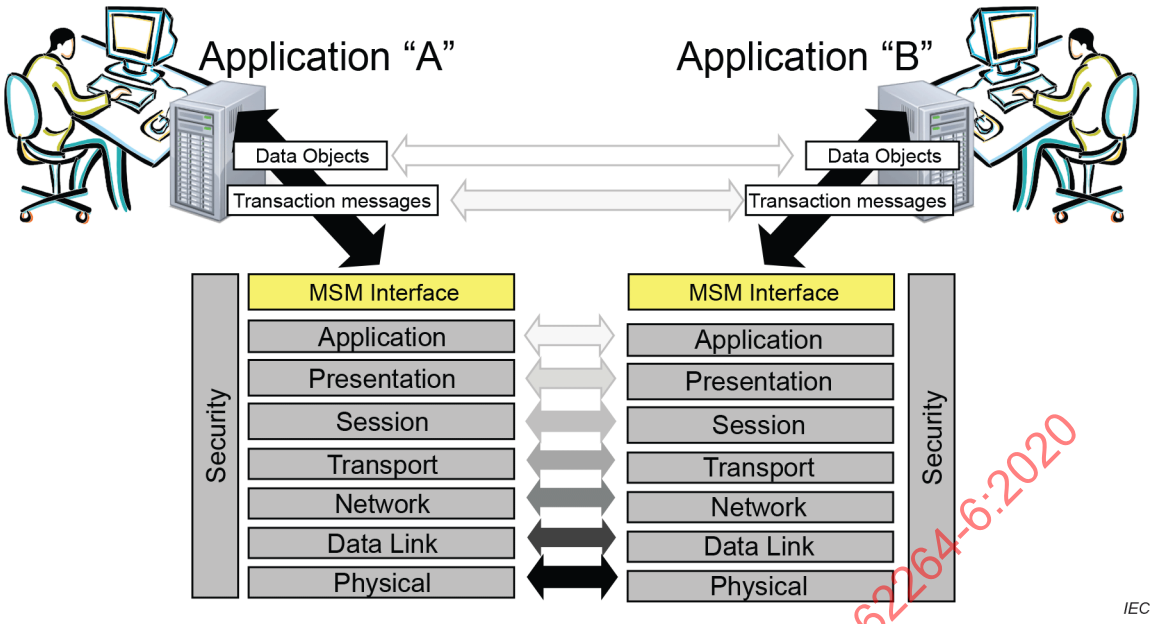


Figure 3 – Positioning and role of MSM

4.2 Abstract service model

The MSM defines a set of common abstract services that is located above the application layer of the OSI model, and that is used for exchanging transaction messages based on the transaction models defined in IEC 62264-5. These services, which are called MSM Service, provide a method for multiple applications to communicate using the transaction models defined in IEC 62264-5. The MSM:

a) does not define how the services are implemented,

b) does not define the architecture of the supporting application,

c) does not define any specific underlying communication method.

The MSM provides an implementation independent common interface to different MSM service providers.

The characteristics of services, such as security features, reliability, guaranteed delivery, quality of service, transformation capability, and other features are not defined in this document.

4.3 Transaction models supported

The MSM defines a standard interface for applications to exchange data using any of the transaction models defined by IEC 62264-5 (i.e. PUSH model, PULL model and PUBLISH model).

Although any transaction models defined by IEC 62264-5 can be exchanged through the MSM service, the following limitations on verbs that can be used for MSM services shall be considered:

a) for a PUBLISH model with multiple subscribers and multiple publishers, the subscribers and publishers can have no direct knowledge of other applications. Only SYNC messages exchanges can be used for this model;

b) for a PUSH model and PULL model, the application sends unsolicited requests for a service and has no direct knowledge of the receiving application that will process the request. Only GET/SHOW, PROCESS/ACKNOWLEDGE. CHANGE/RESPONSE and CANCEL message exchanges can be used for these models.

4.4 Application roles

MSM uses a simpler terminology for application roles than that used in IEC 62264-5. The roles of Information Provider and Information Receiver of IEC 62264-5 are merged into one role called Provider Application. The roles of Information User and Information Sender of IEC 62264-5 are merged into one role called Consumer Application. The difference of the terminology between this document and IEC 62264-5 is shown in Figure 4.

NOTE IEC 62264-5 defines four roles:

1) Information Provider (to receive GET messages and send SYNC messages),
2) Information Receiver (to receive PROCESS, CHANGE, and CANCEL messages),
3) Information Users (to send GET messages and receive SYNC messages),
4) Information Sender (to send PROCESS, CHANGE, and CANCEL messages).

The Provider Application is the owner of data. The Provider Application can publish changes to the data, can receive requests to change the data, and respond to queries for the data. A Consumer Application uses and manipulates data owned by Provider Applications. It can subscribe to changes of the data, can send requests to change the data, and receive reported information.

NOTE The phrase "owner of data" is used to identify the application that has responsibility for enforcing the consistency of data.

An application can be a provider application, consumer application or both.

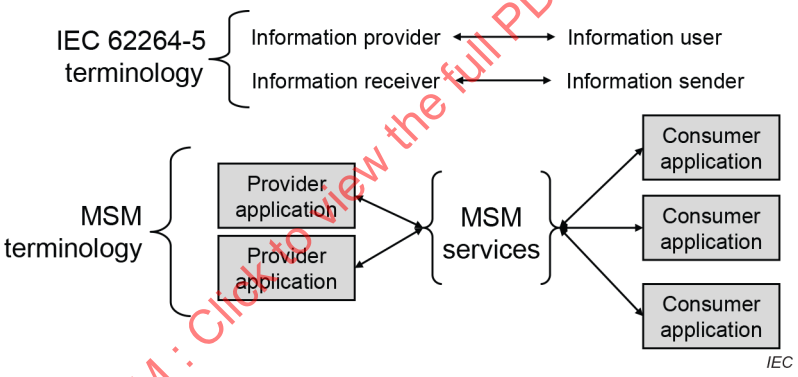


Figure 4 – Messaging service model terminology

4.5 MSM channels

The MSM is based on the abstract concept of MSM channels. An MSM channel represents a specific communication pathway between applications. MSM channels can be of a type used for requests or a type used for publications as shown in Table 1.

Each MSM channel is identified by a URI or equivalent identifier and supports two-way communications between provider applications and consumer applications. An MSM channel is created to support either publication channel services or request channel services. MSM channels have associated topics, which are identified when subscribing to a channel, when posting a publication, and when posting a request.

A provider application posts a publication to a publication channel. A consumer application subscribes to a publication channel and reads publications. Consumer applications could subscribe to publication notifications if supported by the specific publication channel service. If notifications are not supported, then the consumer application polls the publication channel using the read publication service.

A consumer application posts a request to a request channel. A provider application subscribes to a publication channel and reads publications. Provider applications could subscribe to publication notifications if supported by the specific publication channel service. If notifications are not supported, then the provider application polls the publication channel using the read publication service.

Table 1 – Application roles, channels, and services

Application role	Channel type associated with role	MSM services available in the channel
Provider	Publication channel	Provider publication services
Provider	Request channel	Provider request services
Consumer	Publication channel	Consumer publication services
Consumer	Request channel	Consumer request services

4.6 MSM channel services

4.6.1 Types of channel services

MSM defines the following groups of channel service:

– Channel management services,
– Publication channel services,
– Request channel services.

4.6.2 Channel management services

The services provided for each MSM channel are known as the channel management services. The channel management services are used to create and delete channels and to control the Security Token specification for channels.

The channel management services are shown in Figure 5. These services would usually be called by a provider application, or by a dedicated channel management application.

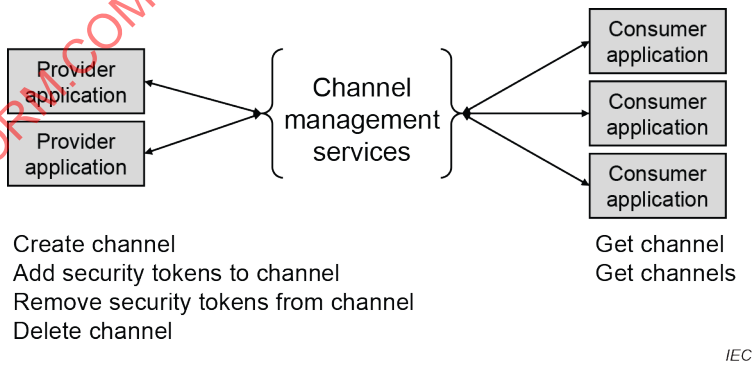


Figure 5 – Channel management services

4.6.3 Publication channel services

The Publication Channel Services are used to post, expire, remove, and read publication messages.

The Publication Channel Services are shown in Figure 6. The services allow multiple Provider Applications to post publications to the same channel. Consumer Applications may subscribe to notifications (if supported by the channel) and may read publications.

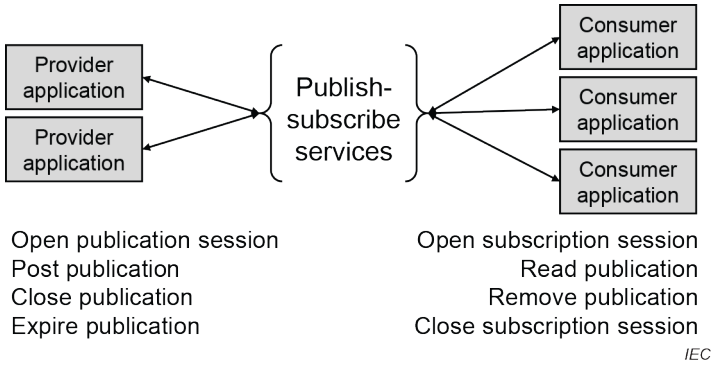


Figure 6 – Publication channel services

4.6.4 Request channel services

The Request Channel Services are used to post request messages and read response messages.

The Request Channel Services are shown in Figure 7. The services allow one or more Consumer Applications to post requests to Provider Applications, allow one or more Provider Applications to read requests and post responses, and for the Consumer Application to read the response. Each posted request includes an additional qualifier, called a "Topic", which allows Provider Applications to determine if the request is relevant to them, and then receive the request and post a response to the requestor.

EXAMPLE Topics define the format and content of a message as identified by the XML Schema Definition (XSD) used to create and verify a message.

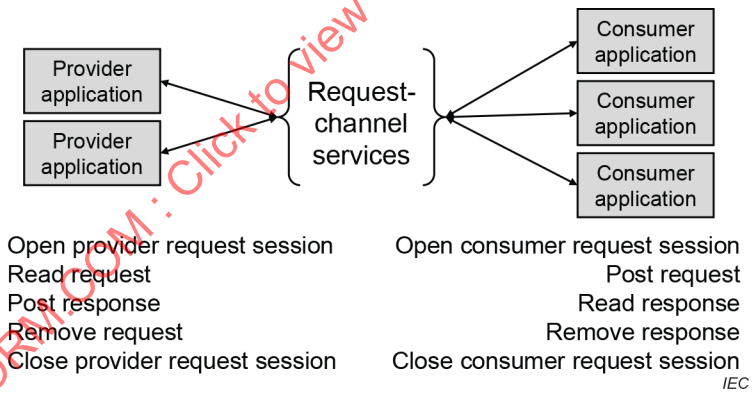


Figure 7 – Request channel services

4.7 Notify listener service

The notify listener service is used as the means to indicate to a provider or consumer application that a message is ready to be read. The notify listener service provides a method for an alternative to polling the MSM service. The Notify Listener Service is shown in Figure 8.

The notify listener service is available on producer and consumer applications after the session is opened with specific listener identification parameters.

The notify listener service is optional for an MSM service provider.

If a provider/consumer application does not provide a Listener Identification for notification, then notification is not provided to the application.

NOTE The format of the notify listener parameter will be defined by the implementation technology.

EXAMPLE A SOAP and Web Service implementation might define the Listener Identification as a valid URL that defines a notify listener service, which is managed by the application creating the session.

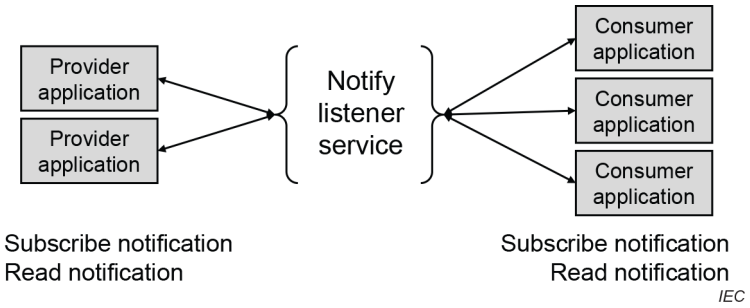


Figure 8 – Notify listener service

5 Methods of operation of channels

5.1 Channel and topic identification

Channels and topics shall be used for filtering messages. Channels identify the scope of the information to be filtered. Topics identify the types of information to be filtered.

This clause defines a recommended method for identifying channels and topics that can be used in order to ensure maximum interoperability.

5.2 Channel names and hierarchy

5.2.1 Channel names

Channel names shall be defined as a name hierarchy using the URI syntax.

5.2.2 Channel name hierarchy

Channel names should follow the naming hierarchy:

\ <MSM root> \ <channel scope> \ <information scope> \ <channel use>

EXAMPLE 1 \AJAXEnterprises\Company\Material\Checkpoint

EXAMPLE 2 \AJAXEnterprises\Company\Material\Request

EXAMPLE 3 \SystemTest\Final\OurMaterialManager\Inventory\Changes

EXAMPLE 4 \AJAXEnterprises\France\Personnel\Checkpoint

5.2.3 MSM root

The MSM root indicates the root of a channel name hierarchy that is defined when the MSM service is installed or initialized. Although every channel name hierarchy shall have one MSM root, those MSM roots may be different from each other.

EXAMPLE 1 An MSM root is the name of the company.
- Such as: "AJAX" or "AJAXEnterprises\SpecialToolCo".

EXAMPLE 2 An MSM root is a related set of services, with sets for testing, deployment, and operations.
- Such as: "SystemTest\Beta", “SystemTest\Final”, "SpecialToolCo\Operations".

NOTE The services do not contain a method to browse the MSM Roots that are defined. These special services can be provided by a service implementation and can have security restrictions that are outside the scope of the MSM services.

5.2.4 Channel scope

The channel scope shall contain a role-based equipment hierarchy, as defined in IEC 62264-1, which could correspond to a physical, geographical, or logical grouping determined by the enterprise, application or project. It can be used to limit the scope of the exchanged information, such as information only exchanged within one division of a company. The channel scopes can include site, area, workcenter, or any other enterprise defined equipment hierarchy element.

EXAMPLE 1 A channel scope can include a site or region name to limit the number of distributed messages, such as: "AsiaPacific", “SouthAfrica”, or “France".

EXAMPLE 2 A channel scope can be a software system, because the information is provided by a well-known system name, such as "OurMaterialManager", “PersonnelTracker”, "InventoryDatabase".

EXAMPLE 3 A channel scope can be company-wide because the information is intended for any application in the company. In this case the channel scope indicates the entire enterprise or company, such as "Enterprise" or "Company”.

5.2.5 Information scope

The information scope defines the range or general type of information exchanged. The information scope may be related to transaction nouns defined in IEC 62264-2 and IEC 62264-4 or related to other collections of objects.

EXAMPLE 1 An application which handles all forms of material information can define a channel with an information scope of "Material".

EXAMPLE 2 An application that only handles Material Lot and Sublot inventories can define a channel with an information scope of "Inventory”.

5.2.6 Channel use

The channel use qualifies the information scope to indicate how the information is being used. The channel use may be related to transaction verbs or other business or control process that deal with how the information on the channel is to be used.

The channel types that shall be used for each of the transaction verbs are specified in Table 2.

Table 2 – Channel use for transaction verbs

Transaction type	Channel type
SYNC ADD	Publication
SYNC CHANGE	Publication
SYNC DELETE	Publication
GET (SHOW)	Request
PROCESS (ACKNOWLEDGE)	Request
CHANGE (RESPOND)	Request
CANCEL	Request

NOTE To support interoperability, channel use can correspond to the classes of transaction message verbs, as defined in IEC 62264-5 or other standards.

EXAMPLE 1

– An application that sends GET messages defines a channel with a channel use of "Query".

– An application that sends PROCESS, CHANGE, and CANCEL messages defines a channel with a channel use of “Command".

– An application that sends SYNC messages defines a channel with a channel use of "Publication". This channel would be used as a publication channel for a snapshot of all of the exchanged information.

EXAMPLE 2 Publication Changes and Publication Checkpoint channel are used together by a provider application as shown in Figure 9.

– The Checkpoint channel would be used to publish a current snapshot of all of the exchanged information.

– The Changes channel would be used to publish all changes since the last snapshot.

After a checkpoint is published, the provider application would clear all publications from the Changes channel and clear previous Checkpoint publications.

This dual publication channel method allows a consumer application to quickly sync all published information on a topic, without excessive interaction with the MSM service.

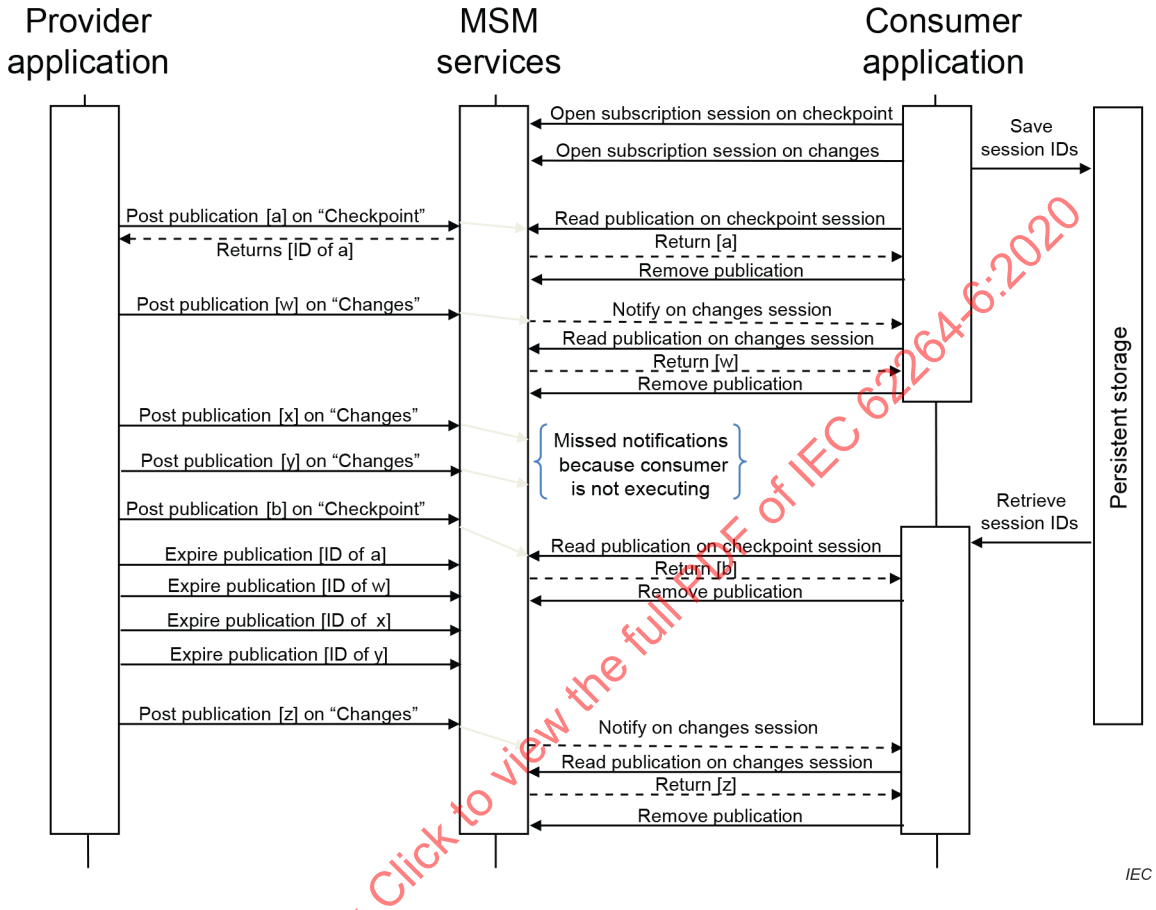


Figure 9 – Changes and checkpoint channel example

5.3 Publication expiration

Information sent using a publication channel might only be valid for a certain period of time after which it expires or can be marked as expired by the publication service. Expired publications shall not be available to subscribing consumer applications nor accessible to the provider application. If an already read message subsequently expires, it is still available to the consumer to ensure that a Remove Publication call removes the correct message.

A publication can be flagged as expired by a provider application via the Expire Publication service or through an expiration time defined when the publication is posted.

With the time-based expiration, the expiration time is calculated based on the completion of invocation of the Post Publication service plus the specified duration.

Any publication message can be expired through the Expire Publication service.

5.4 Topics

5.4.1 Topic definition

Topics are used by application services to limit or filter the type of information that is obtained from the channel for provider applications and consumer applications.

Topics are used by provider applications to specify the type of information that they will be publishing or posting on an MSM channel.

Topics allow a single channel to handle a collection of different data, yet still provide a method for the receiver of the data to limit the types of data that it is required to handle.

The same topic may be used across multiple channels.

EXAMPLE 1 There might be a ProductionSchedule topic defined for CheckPoint and Changes channels with a site channel scope, and a ProductionSchedule topic used for Checkpoint and Changes channels for an area channel scope.

EXAMPLE 2 There might be a QualificationTest topic used for a Request channel at the enterprise channel scope, and a QualificationTest topic used for a Request channel at the country channel scope.

5.4.2 Topic names

To support interoperability of IEC 62264 implementations, the topic names shall correspond to an identification of the implementation of the transaction message verbs and nouns, as defined in IEC 62264-5.

EXAMPLE 1 Classes of nouns from IEC 62264-5

Equipment Class	Equipment	Capability Test
Personnel Class	Person	Qualification Test
Material Class	Material Definition	Material Lot
Material Sublot	Material Test	
Operations Capability	Operations Definition	Operations Performance
Operations Schedule	Process Segment	Production Capability
Product Definition	Production Schedule	Production Performance
Resource Relationship Network	Transaction Profile	Work Alert
Work Capability	Work Definition	Work Performance
Work Schedule	Workflow Specification	

The topic names shall contain the associated standard and version number of the associated standard or noun.

EXAMPLE 2 One topic might be defined for messages using B2MML-V0402-MaterialLot definitions, another for B2MML-V0501-MaterialLot definitions and a third topic for messages using B2MML-V0600-MaterialLot definitions.

The same topic may be defined on multiple channels.

EXAMPLE 3 There might be a ProductionSchedule topic defined for CheckPoint and Changes channels with a site channel scope, and a ProductionSchedule topic defined for Checkpoint and Changes channels for an area channel scope.

EXAMPLE 4 There might be a QualificationTest topic defined for a Request channel at the enterprise channel scope, and a QualificationTest topic defined for a Request channel at the country channel scope.

5.5 Sessions

All communication over any channels are exchanged through a session. Sessions are created using open session services (i.e. Open publication session, Open subscription session, Open provider request session, and Open consumer request session), which return a session ID.

Session IDs shall be persistent and shall not be tied to the execution instance of the requesting program. A session ID remains valid even if the calling program stops and restarts, as long as the calling program maintains the session ID (in storage) and can read it on restart, then the MSM session is still available. See 7.1.2 for an example of reusing a session ID after restart of an application.

5.6 Security

5.6.1 Secure message exchanges

Security measures for message exchanges shall have authenticated access to channel management services.

NOTE Security in the MSM services is of paramount importance. In the MSM Service model, the communication applications have no knowledge of their communication partners, and do not know if there are none (for a publisher with no subscriptions), one, or many. Therefore, security cannot be defined as communication with trusted partners, but as communication through secure channels.

5.6.2 Security tokens on channels

Channel access security shall be managed through security tokens.

Security tokens are assigned to channels.

Security tokens on channels may be added to channels by the users of the MSM service.

NOTE While the MSM service provider is required to provide security, the final users of the MSM service might decide not to assign any security tokens to one or more channels, in which case the channels might be accessed without any authentication control.

Security tokens shall be used by applications when opening or subscribing to a channel. If the application provided Security Token does not match a Security Token assigned to the channel, then no channel information is returned.

Security tokens are exchanged in an out-of-band communication channel, such as manual exchange of tokens, or electronic exchange through a secure point-to-point channel, as shown in Figure 10.

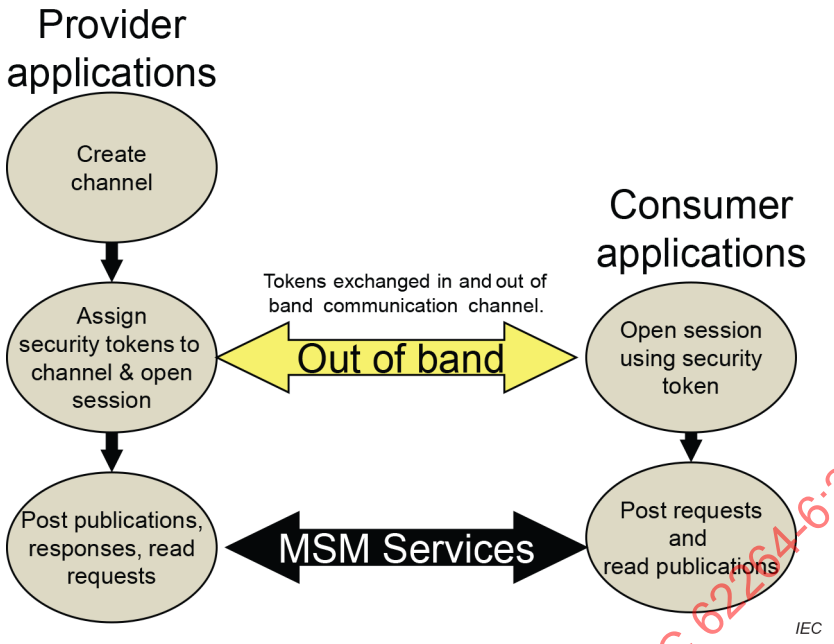


Figure 10 – Security of channels

5.6.3 Security token format

The Security Token format shall be defined in the MSM implementation specification. Different implementations may have different methods and formats for the security tokens.

5.6.4 MSM service provider implementations

a) All MSM Service Providers shall implement security tokens.

b) MSM Service Providers may limit the ability to use the Channel Management services to approved applications or users in order to increase security.

c) In an implementation, providers and consumers shall arbitrate the level of security to be used for a channel, if any. While there is a requirement that the services provide security services, there is no requirement that a specific implementation use the services.

EXAMPLE 1 A system might share information across companies through the public Internet. In this case, an MSM Service Provider implementation provides a strong Security Token system through a public key mechanism with specific Security Token assigned to specific communicating companies.

EXAMPLE 2 A system can be entirely contained within a secure environment behind both corporate and operations firewalls. In this case, the user might decide to not assign security tokens to channels and rely on other measures to ensure security.

6 Service definitions

6.1 Type definitions

Table 3 contains the type definitions that are associated with the service definitions.

Table 3 – Type definitions

Type	Description
Channel Description	Text used in browsing channels and to provide assistance in maintenance of an MSM implementation
Channel Type	Indicates whether the channel is for publications, requests or responses. The MSM implementation may use the channel type to ensure that the correct session creation service is called for a channel. Defined Channel Types are "Publication" and "Request".
Channel URI	The primary identifier for a channel. The Channel URI should be a wide string allowing channel names with international character sets. See 5.2 for details on the format.
Filter Expression	An optional filtering element that may be applied to messages on a channel. The format of the filter expression is not defined in this document, but would be defined in an MSM implementation specification.
Session ID	An identifier generated by the MSM upon an application creating a session on a channel and provided to the application for use in the MSM service.
Listener Identification	An implementation defined element that is used to indicate to an application when a new message has arrived. EXAMPLE A URI endpoint, reachable by the MSM Service Provider
Message ID	An identifier generated by the MSM upon posting of a message to a channel in a session
Message Content	The message content
Message Expiry	The duration until the expiration of a publication message on a publication channel
Security token	The security tokens assigned to the channel
Session ID	A unique identification of the communication session used in using an MSM channel
Topic	An identification of the topic of a message, see 5.4

6.2 Defined return value of services

Table 4 contains the pre-defined return values of MSM service for specific faults. These types include a fault indicator that is represented by text, numeric code or enumeration. When a return value for a fault is defined, a human readable explanation should be included in its description.

Table 4 – Service fault definitions

Type	Description
Channel Fault	Error returned when a Channel URI is invalid, or the application does not have the appropriate Security Token to access the channel
Operation Fault	Error returned when an illegal operation for the channel type is attempted
Parameter Fault	Error indicating a missing or invalid parameter passed to a service
Security Token Fault	Error returned when an invalid Security Token is used
Session Fault	Error returned when an invalid Session ID is used in a service

Table 5 contains the service parameter definitions.

Table 5 – Service parameter definitions

Type	Description
Listener Identification	An identification of a listener function, defined by the implementation technology
Message ID	A unique identification of a message
Publication Message	A message which should be in the format defined by IEC 62264-5 SYNC messages
Request Message	A message which should be in the format defined by IEC 62264-5 GET, PROCESS, CHANGE, and CANCEL messages
Request Message ID	The Message ID of a request message
Response Message	A message which should be in the format defined by IEC 62264-5 SHOW, ACKNOWLEDGE, RESPOND, and CONFIRM messages

6.3 Channel management services

6.3.1 Create channel

The create channel service shall have the function, input parameters and returns defined in Table 6.

Table 6 – Create channel

Name	Create channel
Function	Creates an MSM channel of the specified Channel Type If the channel already exists, then a Channel Fault is returned. Any specified security tokens are added to the channel.
Input Parameters	– Channel URI – Channel Type (Publication or Request) – Optional Description of the channel – Optional list of Security Tokens
Returns	– Success or error criteria

6.3.2 Add security tokens

The add security tokens service shall have the function, input parameters and returns defined in Table 7.

Table 7 – Add security tokens

Name	Add security tokens
Function	Adds security tokens to a channel. If the Channel URI does not exist, or if the specified channel is assigned security tokens and the Security Token does not match a token already assigned to the specified channel, then a Channel Fault is returned. If a new Security Token has already been assigned to the channel, then no action is taken.
Input Parameters	– Channel URI – Optional Security Token – List of Security Tokens to add
Returns	– Success or error criteria

6.3.3 Remove security tokens

The remove security tokens service shall have the function, input parameters and returns defined in Table 8.

Table 8 – Remove security tokens

Name	Remove security tokens
Function	Removes security tokens from a channel. If the Channel URI does not exist, then a Channel Fault is returned. If the specified channel is assigned security tokens and the Security Token does not match a token assigned to the specified channel, then a Channel Fault is returned. If any Security Token to be removed is not assigned to the channel, then a Security Token Fault is returned and no tokens are removed from the channel, even if they are valid.
Input Parameters	– Channel URI – Security Token – List of Security Tokens to remove
Returns	– Success or error criteria

6.3.4 Delete channel

The delete channel service shall have the function, input parameters and returns defined in Table 9.

Table 9 – Delete channel

Name	Delete channel
Function	Deletes an MSM Channel. If the Channel URI does not exist, then a Channel Fault is returned. If the specified channel is assigned security tokens and the Security Token does not match a token assigned to the specified channel, then a Channel Fault is returned. The channel along with associated sessions and messages are no longer available. No notification is provided to any applications with active sessions.
Input Parameters	– Channel URI – Optional Security Token
Returns	– Success or error criteria

6.3.5 Get channel

The get channel service shall have the function, input parameters and returns defined in Table 10.

Table 10 – Get channel

Name	Get channel
Function	Gets information about a channel. If the Channel URI does not exist, then a Channel Fault is returned. If the specified channel is assigned security tokens and the Security Token does not match a token assigned to the specified channel, then a Channel Fault is returned.
Input Parameters	– Channel URI – Optional Security Token
Returns	– Success or error criteria – Channel URI – Channel Type – Channel Description

6.3.6 Get channels

The get channels service shall have the function, input parameters and returns defined in Table 11.

Table 11 – Get channels

Name	Get channels
Function	Gets information about all channels where the Security Token matches one of the channel's Security Tokens. If there is no match, then an empty list of channel URIs is returned.
Input Parameters	– Optional Security Token
Returns	– Channel URIs – Channel Types – Channel Descriptions

6.4 Notify listener service

The notify listener service shall have the function, input parameters and returns defined in Table 12.

Table 12 – Notify listener

Name	Notify listener
Function	Receives a notification of a new message on a channel, using the Listener Identification specified when a session is opened. This is a function provided by the provider or consumer application when subscribing to a channel on which notifications can be received. The interface is defined so that the applications can define their local functions with the correct interface for notifications.
Input Parameters	Input parameters are defined by the specific MSM technology implementation.
Returns	Returns are defined by the specific MSM technology implementation.

NOTE Notify listener services will generally return a SessionID, MessageID, Topic, and optional RequestMessageID.

6.5 Provider publication services

6.5.1 Open publication session

The open publication session service shall have the function, input parameters and returns defined in Table 13.

Table 13 – Open publication session

Name	Open publication session
Function	Opens a publication session for a channel and returns an ID for the session. If the Channel URI does not exist, then a Channel Fault is returned. If the specified channel is assigned security tokens and the Security Token does not match a token assigned to the specified channel, then a Channel Fault is returned. If the channel type is not a Publication type, then an Operation Fault is returned.
Input Parameters	– Channel URI – Optional Security Token
Returns	– Success or error criteria – Session ID

6.5.2 Post publication

The post publication service shall have the function, input parameters and returns defined in Table 14.

Table 14 – Post publication

Name	Post publication
Function	Posts a publication message to a channel and creates a message with the Message Content and a Message ID that uniquely identifies message, and makes it available for subscribers. If the Session ID does not exist or does not correspond to a publication session, then a Session Fault is returned. If a Topic is blank, then a Parameter Fault is returned.
Input Parameters	– Session ID – Publication Message – List of topics – Optional Expiration Duration for the publication
Returns	– Success or error criteria – Message ID of posted message

6.5.3 Expire publication

The expire publication service shall have the function, input parameters and returns defined in Table 15.

Table 15 – Expire publication

Name	Expire publication
Function	The publication is no longer available to subscribers. If an already read message subsequently expires, it is still available to the consumer to ensure that a RemovePublication call removes the correct message. If the Session ID does not exist or does not correspond to a publication session, then a Session Fault is returned. If the Message ID does not correspond with the Session ID or the corresponding message has already expired, then no action is taken. The message is expired for all topics associated with the message.
Input Parameters	– Session ID – Message ID
Returns	– Success or error criteria

6.5.4 Close publication session

The close publication session service shall have the function, input parameters and returns defined in Table 16.

Table 16 – Close publication session

Name	Close publication session
Function	Closes a publication session. All unexpired messages that have been posted during the session will be expired. If the Session ID does not exist (non-existent or already closed), then a Session Fault is returned. If the Session ID does not correspond to a Publication channel type, then a Session Fault is returned.
Input Parameters	– Session ID
Returns	– Success or error criteria

6.6 Consumer publication services

6.6.1 Open subscription session

The open subscription session service shall have the function, input parameters and returns defined in Table 17.

Table 17 – Open subscription session

Name	Open subscription session
Function	Open a subscription session for a channel. A subscription session will not pick up messages posted prior to the subscription session start. If the Channel URI does not exist, then a Channel Fault is returned. If the specified channel is assigned security tokens and the specified Security Token does not match a token assigned to the specified channel, then a Channel Fault is returned. If the channel type is not a Publication type, then an Invalid Operation Fault is returned. If a Topic is blank, then an Invalid Parameter Fault is returned.
Input Parameters	– Channel URI – List of Topics (subscribed to) – Optional Security token – Optional Listener Identification – Optional Filter Expression
Returns	– Success or error criteria – Session ID

6.6.2 Read publication

The read publication service shall have the function, input parameters and returns defined in Table 18.

Table 18 – Read publication

Name	Read publication
Function	Returns the first non-expired publication message (if any) that satisfies the session's topics. If the Session ID does not exist, then a Session Fault is returned. If the Session ID does not correspond to a publication session, then a Session Fault is returned. If there are no publication messages, then a null publication message is returned.
Input Parameters	– Session ID
Returns	– Success or error criteria – Publication Message – Message ID – List of topics for the Publication Message

6.6.3 Remove publication

The remove publication service shall have the function, input parameters and returns defined in Table 19.

Table 19 – Remove publication

Name	Remove publication
Function	Removes the first publication message in the subscription queue. If there is no publication message, then no action is taken. If the Session ID does not exist, then a Session Fault is returned. If the Session ID does not correspond to a publication session, then a Session Fault is returned.
Input Parameters	– Session ID
Returns	– Success or error criteria

6.6.4 Close subscription session

The close subscription session service shall have the function, input parameters and returns defined in Table 20.

Table 20 – Close subscription session

Name	Close Subscription Session
Function	Close the subscription session. If a notify listener service has been requested, then there will be no more notifications. Any unread publications for the session are no longer available. If the Session ID does not exist, then a Session Fault is returned. If the Session ID does not correspond to a publication session, then a Session Fault is returned.
Input Parameters	– Session ID
Returns	– Success or error criteria

6.7 Provider request services

6.7.1 Open provider request session

The open provider request session service shall have the function, input parameters and returns defined in Table 21.

Table 21 – Open provider request session

Name	Open Provider Request Session
Function	Opens a provider request session for reading requests and posting responses. If the Channel URI does not exist, then a Channel Fault is returned. If the specified channel is assigned security tokens and the specified Security Token does not match a token assigned to the specified channel, then a Channel Fault is returned. If the Channel Type is not a Request type, then an Invalid Operation Fault is returned. If a Topic is blank, then an Invalid Parameter Fault is returned.
Input Parameters	– Channel URI – List of Topics – Optional Security token – Optional Listener Identification – Optional Filter Expression
Returns	– Success or error criteria – Session ID

6.7.2 Read request

The read request service shall have the function, input parameters and returns defined in Table 22.

Table 22 – Read request

Name	Read request
Function	Returns the first request message in the message queue for the session. This service does not remove the message from the message queue. The returned Topic will correspond to the topic that matched the posted request. If the Session ID does not correspond to a provider request session, then a Session Fault is returned.
Input Parameters	– Session ID
Returns	– Success or error criteria – Returned Message – Returned Message ID – Topic of the returned message

6.7.3 Remove request

The remove request service shall have the function, input parameters and returns defined in Table 23.

Table 23 – Remove request

Name	Remove request
Function	Removes the first request message from the request session. If the Session ID does not exist, then a Session Fault is returned. If the Session ID does not correspond to a provider Request session, then a Session Fault is returned.
Input Parameters	– Session ID
Returns	– Success or error criteria

6.7.4 Post response

The post response service shall have the function, input parameters and returns defined in Table 24.

Table 24 – Post response

Name	Post response
Function	Removes the first request message from the request session. If the Session ID does not exist, then a Session Fault is returned. If the Session ID does not correspond to a provider Request session, then a Session Fault is returned.
Input Parameters	– Session ID – Response Message
Returns	– Success or error criteria

6.7.5 Close provider request session

The close provider request session service shall have the function, input parameters and returns defined in Table 25.

Table 25 – Close provider request session

Name	Close provider request session
Function	Closes a provider request session. If the Session ID does not exist, then a Session Fault is returned. If the Session ID does not correspond to a provider request session, then a Session Fault is returned.
Input Parameters	– Session ID
Returns	– Success or error criteria

6.8 Consumer request services

6.8.1 Open consumer request session

The open consumer request session service shall have the function, input parameters and returns defined in Table 26.

Table 26 – Open consumer request session

Name	Open consumer request session
Function	Open a consumer request session for posting requests and reading responses. If the Channel URI does not exist, then a Channel Fault is returned. If the specified channel is assigned security tokens and the specified Security Token does not match a token assigned to the specified channel, then a Channel Fault is returned. If the channel type is not a Request type, then an Invalid Operation Fault is returned.
Input Parameters	– Channel URI – Optional Security token – Optional Listener Identification
Returns	– Success or error criteria – Session ID

6.8.2 Post request

The post request service shall have the function, input parameters and returns defined in Table 27.

Table 27 – Post request

Name	Post request
Function	Post a request message on the request channel and return the ID of the message. If the Session ID does not exist, then an Invalid Operation Fault is returned. If the Session ID does not correspond to a consumer request session, then a Session Fault is returned. If a Topic is blank, then an Invalid Parameter Fault is returned.
Input Parameters	– Session ID – Request Message – Topic of the request – Optional Request Timeout
Returns	– Success or error criteria – Request Message ID – Optional Expiration Duration for the request

NOTE Topics are coordinated as part of the system installation. If any system is posting requests that have no providers, then the messages will not be answered. Implementations can consider adding timeouts and return errors if a provider does not pick up a request within a reasonable time.

6.8.3 Read response

The read response service shall have the function, input parameters and returns defined in Table 28.

Table 28 – Read response

Name	Read response
Function	Read the response message from a posted request. If the Session ID does not exist, then a Session Fault is returned. If the Session ID does not correspond to a consumer request session, then an Invalid Operation Fault is returned. If the Request Message ID does not correspond to a message in the message queue, then no message is returned.
Input Parameters	– Session ID – Request Message ID
Returns	– Success or error criteria – Response Message

6.8.4 Remove response

The remove response service shall have the function, input parameters and returns defined in Table 29.

Table 29 – Remove response

Name	Remove response
Function	Remove a response message from the request channel. If the Session ID does not exist, then a Session Fault is returned. If the Session ID does not correspond to a consumer request session, then an Invalid Operation Fault is returned. If the Request Message ID does not correspond to a message in the message queue, then no action is taken
Input Parameters	– Session ID – Request Message ID
Returns	– Success or error criteria

6.8.5 Close consumer request session

The close consumer request session service shall have the function, input parameters and returns defined in Table 30.

Table 30 – Close consumer request session

Name	Close consumer request session
Function	Remove a response message from the request channel. Close a consumer request session. If the Session ID does not exist, then a Session Fault is returned. If the Session ID does not correspond to a consumer request session, then a Session Fault is returned.
Input Parameters	– Session ID
Returns	– Success or error criteria

7 Scenarios

7.1 Publish-subscribe scenarios

7.1.1 Simple publish-subscribe scenario

A simple publish-subscribe scenario with a single provider application and a single consumer application using a notify listener service, is shown in Figure 11.

NOTE 1 There will usually be multiple consumer applications receiving publications, but only one is shown in this example.

NOTE 2 It is assumed that the appropriate channels and topics have been created prior to the scenario.

In this scenario, the provider application opens a publication channel with a channel URI and security token. When the provider application has determined that data should be published, it posts publications (using SYNC messages) with a message topic.

A consumer application subscribes to the publication channel using a channel URI, security token, and list of topics. The session ID is saved in case the consumer application stops unexpectedly.

When a new message with the right topic is posted, the consumer application is notified of the posting and then reads the new publication message from the publication channel.

When the consumer application no longer needs data, it unsubscribes from the subscription session and clears the session ID in persistent storage so that it will open a new session when restarted.

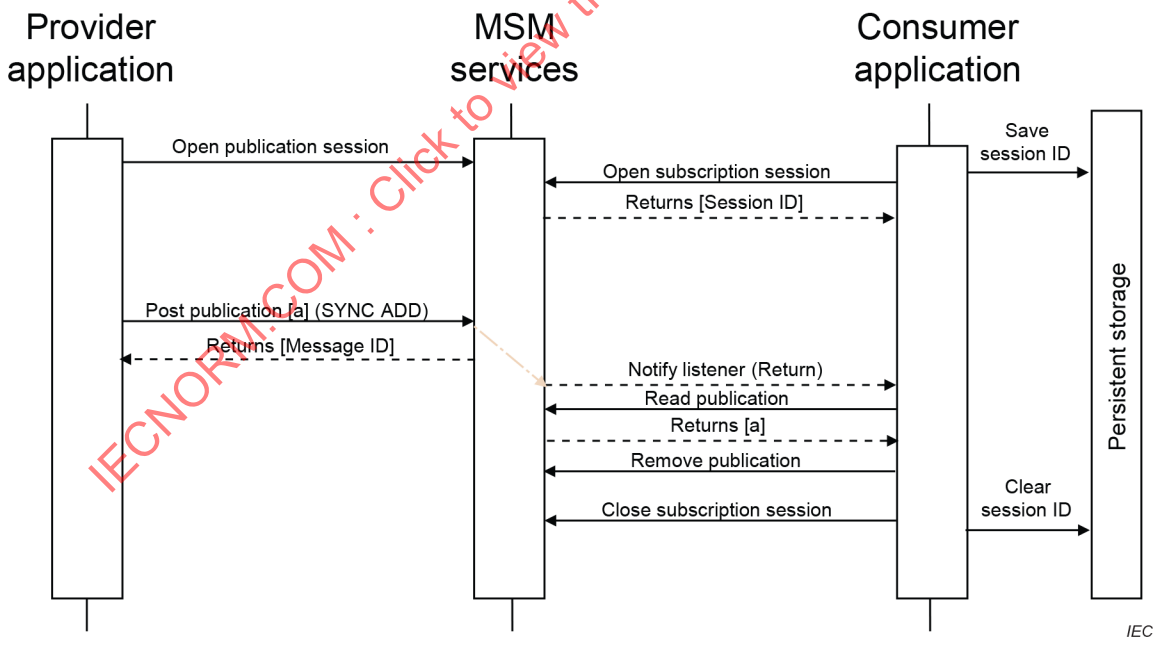


Figure 11 – Publication scenario with notification

7.1.2 Publish-subscribe scenario with multiple messages

A simple publish-subscribe scenario with a single provider application, notification model available, and consumer applications using a notify listener service, for multiple messages is shown in Figure 12.

NOTE See 4.7 for a recommendation of the structure of channels for a more robust actual implementation.

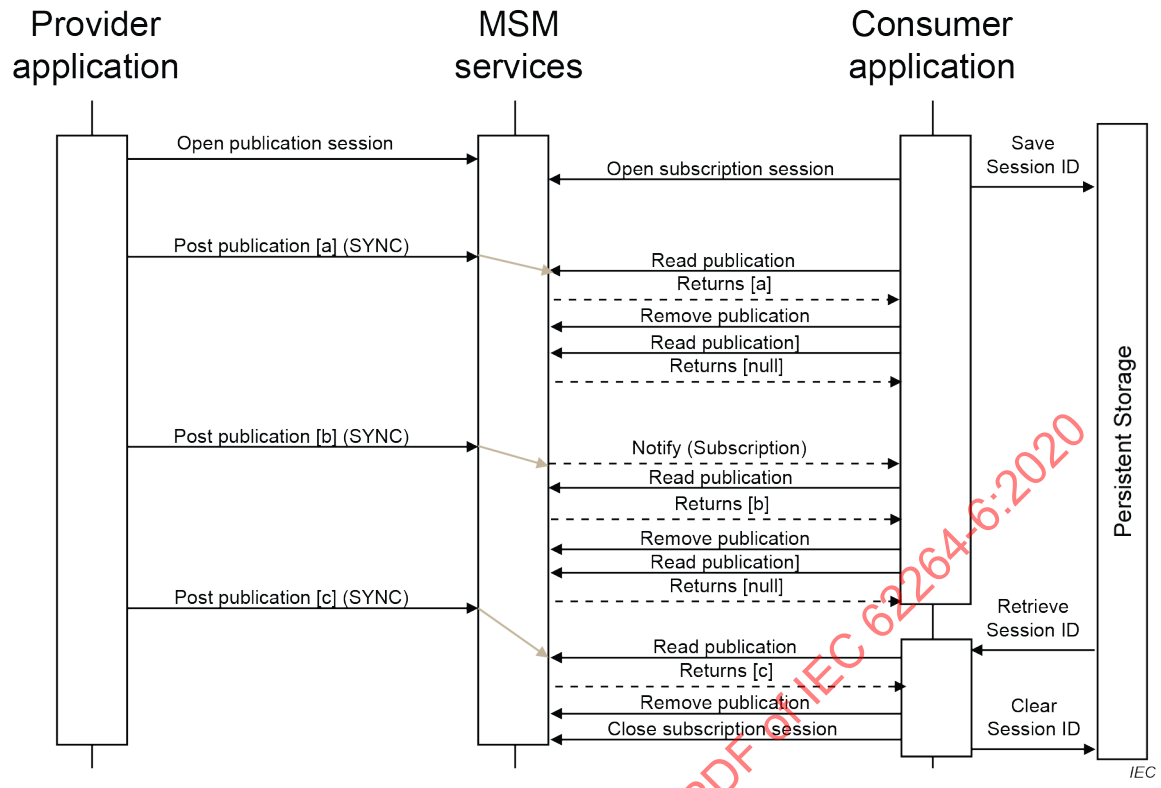


Figure 12 – Publication scenario with multiple messages

In this scenario, the provider application opens a publication session for a given channel. When the provider application has determined that data should be published, it posts publication [a] with a message topic and no message expiry.

A consumer application opens a session to the publication channel using a list of topics. The session ID is stored for later use when the consumer application stops and restarts.

When a notification is received, the consumer application reads and removes all publications until a null is returned from the Read Publication.

When the consumer application stops and restarts, it retrieves the saved session ID. Because the consumer application was not active, there might have been a missed notification, so the application reads and removes all new publications until a null is returned from the Read Publication.

When the consumer application has finished all processing and no longer wants to receive subscriptions, it closes the subscription session and clears the saved session ID.

7.1.3 Publish-subscribe scenario without notification

A publish-subscribe scenario with a single provider application, where notification is not available or the consumer application is not able to use notification, is shown in Figure 13. In this scenario, there is no change for the actions of the provider application from the scenario in 7.1.1.

In this scenario, the consumer application would poll the publication channel for publications either periodically or based on some local event. The returned information from the read indicates if a new publication was returned.

The next Read Publication call returns either the next publication from the subscription queue or null if there are no more publications available.

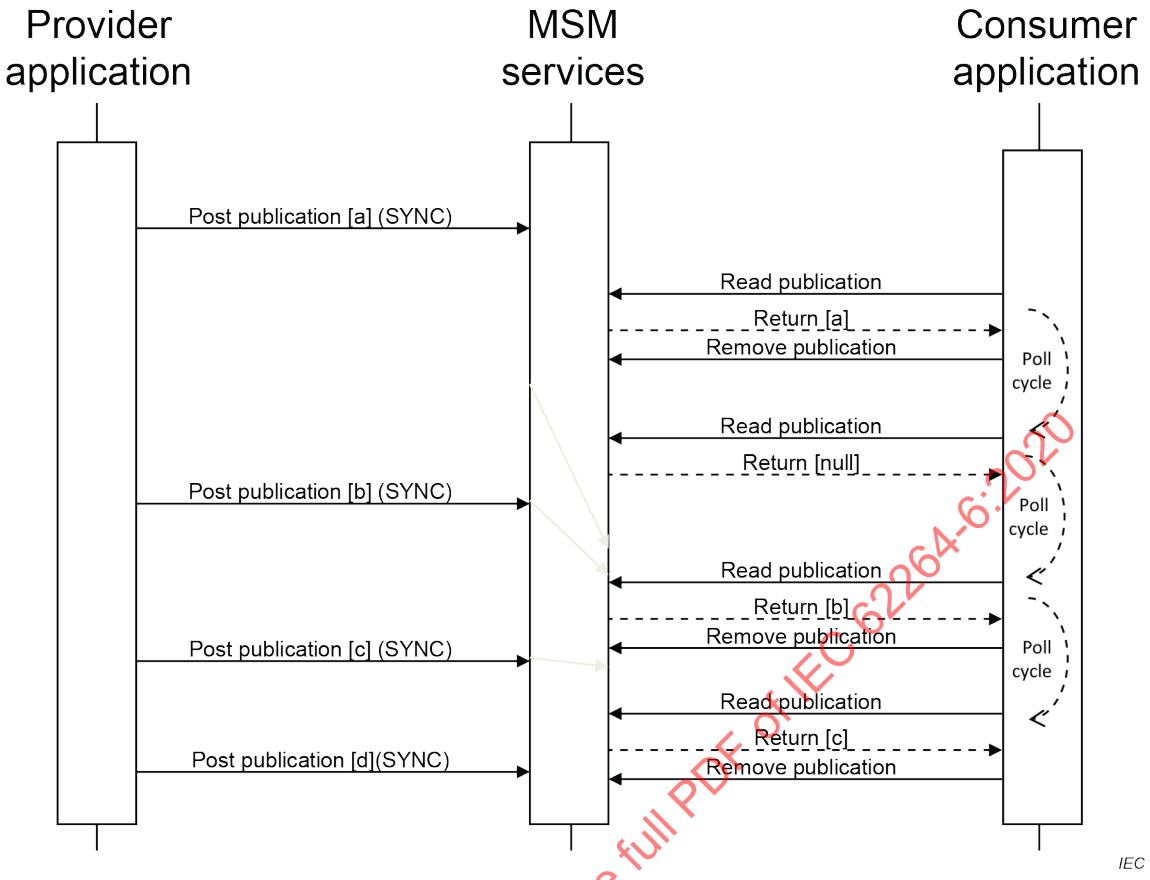


Figure 13 – Publication without notification

7.1.4 Multiple publisher scenario

More than one provider application may use the same publication channel. The scenario shown in Figure 14 has two provider applications. For example, one application could publish changes with topics for Material Definitions while another may publish changes with topics for Material Lots.

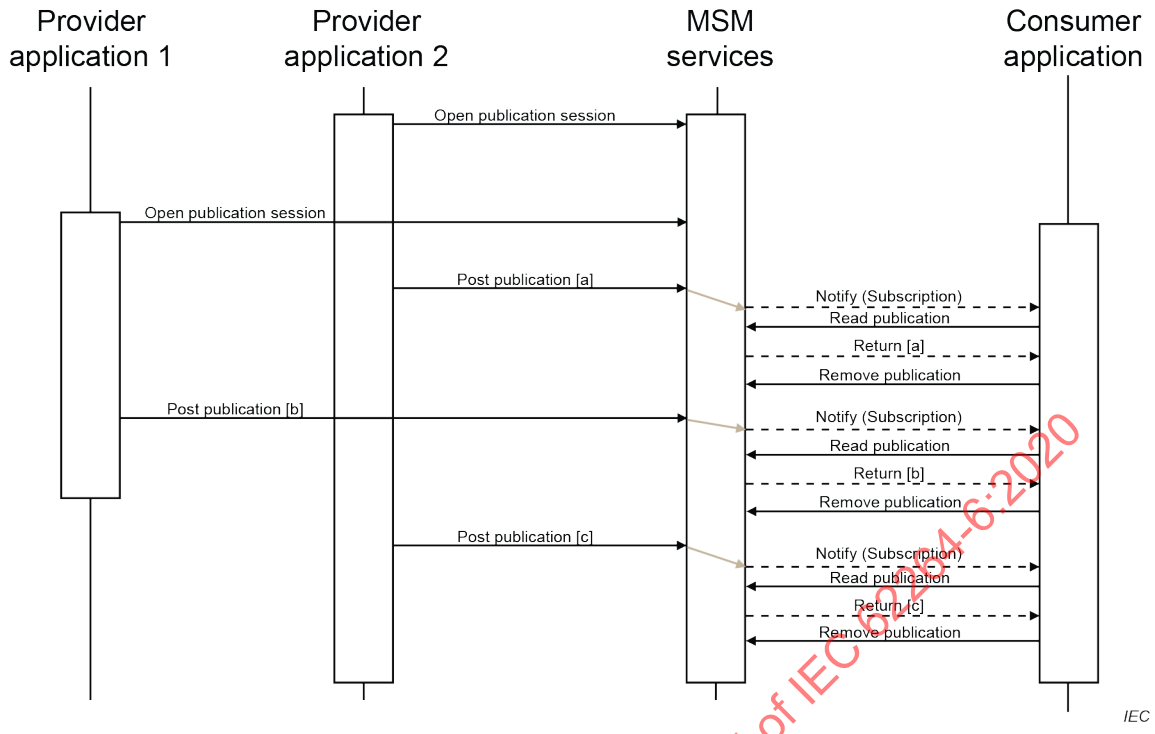


Figure 14 – Publication with multiple provider applications

7.1.5 Publish-subscribe scenario with publication expiration

Message expiration can be used by provider applications to remove messages from a consumer application's visibility. This could be due to a number of reasons, including changing relevancy of the message or inaccuracies in the message. The scenario in Figure 15 highlights expiration behaviour in two cases:

a) where a read message has expired, and

b) where an unread message has expired.

On the second Read Publication call by the consumer application, the MSM Service Provider returns the next message in the subscription queue – although in this case, there is no message. The third Read Publication returns the next unexpired publication in the subscription queue, message [c].

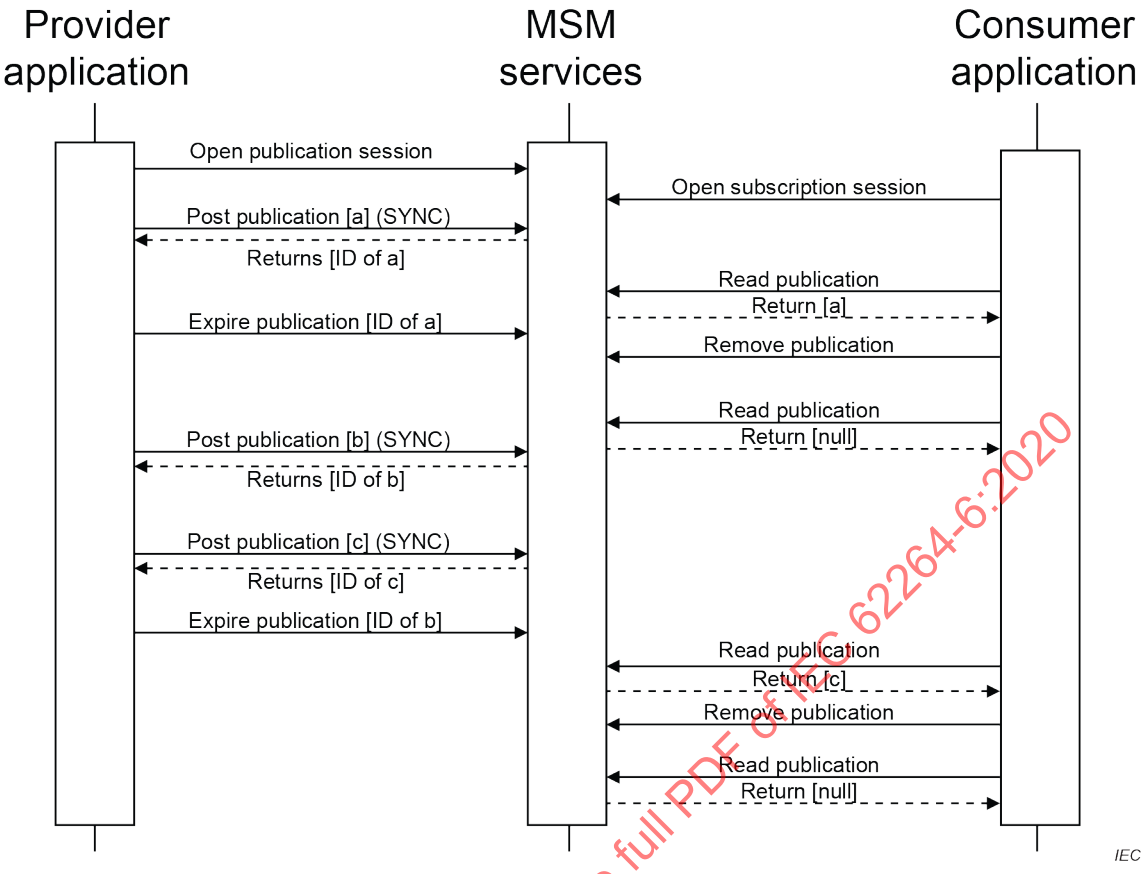


Figure 15 – Publication with expired publications

7.2 Request channel scenarios

7.2.1 Request-response scenario with notification

Figure 16 illustrates a scenario for a GET/SHOW transaction with the provider application, consumer application, and a channel supporting notification.

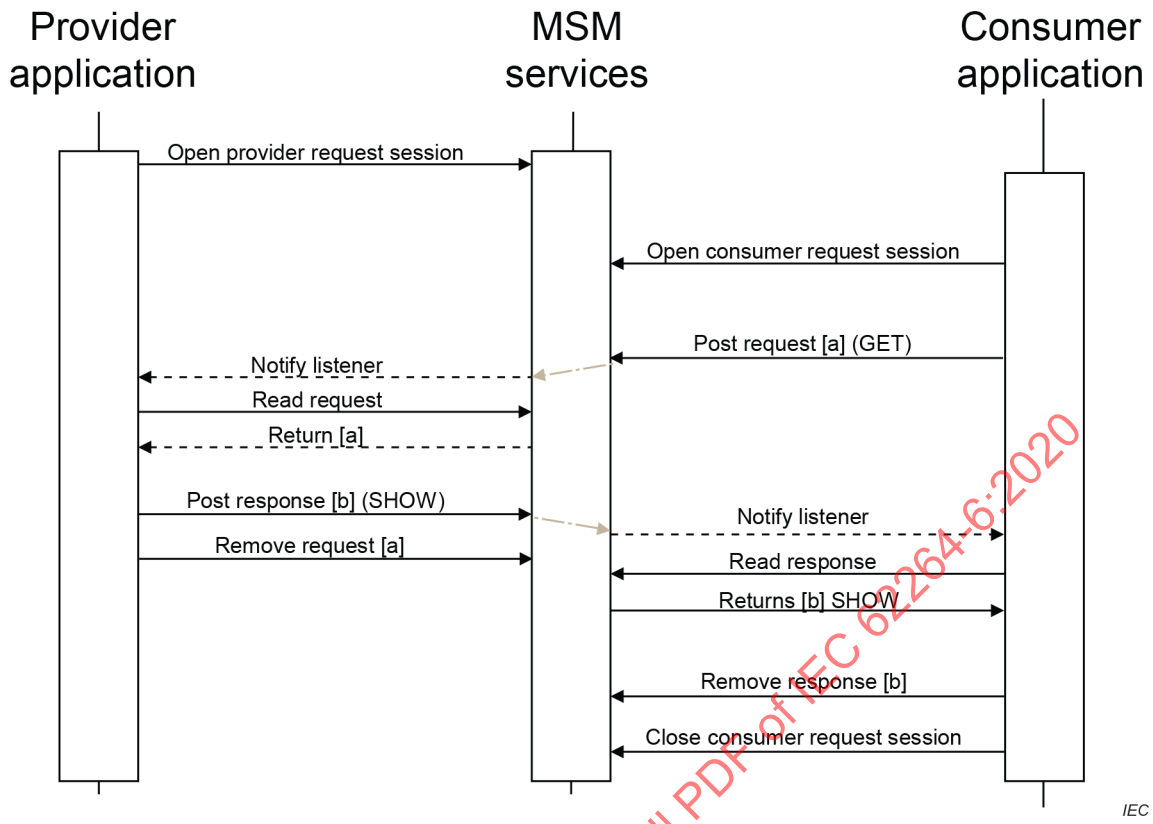


Figure 16 – GET/SHOW request service scenario

In Figure 16, a provider application opens a provider request session. A consumer application opens a consumer request session and posts a request [a]. The provider is notified and reads the request [a]. The provider application performs its appropriate function (in this case to get data) and sends the response message [b] (in this case a SHOW message) and then removes the request [a]. The consumer application is notified of the posting and reads the response [b] and then removes the response [b].

7.2.2 Request-response scenario without notification

If the applications or MSM services do not support notification, then the provider may poll for a request and consumer applications may poll for a response. Figure 17 illustrates a scenario using a CHANGE-RESPONSE transaction where the consumers and providers must poll for a response.